Modeling Fluid Flow in Domains Containing Moving Interfaces

David B. Carrington, T-3;
Dominic A. Muñoz,
Juan C. Heinrich, University of New Mexico

Numerical simulations of fluid flow in domains containing moving rigid objects or boundaries are still challenging when meshes have to be adapted and periodically regenerated. A numerical strategy that falls into the general category of Arbitrary Lagrangian–Eulerian (ALE) methods is being developed. The method is based on a fixed-mesh that is locally modified both in space and time to describe the moving interfaces that are allowed to move independently of the mesh. This results in a fully robust formulation capable of calculating in irregular meshes that contain moving devices of complex geometry and free of mesh entanglement problems. The present work constitutes the first stage in the development of a 3D model to interface with the new KIVA-hpFE simulator. The method's accuracy has been assessed in 2D using a case that has an analytical solution.

The design of internal combustion engines presents significant challenges to the optimization of shape, size, efficiency, power output, environmental impact, etc. Numerical simulations have provided an excellent tool for analysis prior to prototype building and have been used in the design of internal combustion engines for some time now [1,2]. The numerical models used to simulate flow in domains that physically change with time are for the most part based on ALE methods [3,4]. These methods, based on finite differences or finite elements formulations, are combined with moving mesh schemes in which the mesh is deformed or regenerated as the domain evolves to adapt to the changing geometry [5,6].

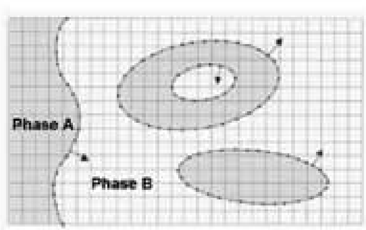

Fig. 1. Schematic of a rectangular domain discretized with a uniform mesh that contains a fluid phase (B) and three types of moving interfaces defined by marker points. The arrows indicate the positive vector normal to the interface.

When the meshes are adapted to fit the evolving geometry they become degraded to the point where they must be regenerated, an expensive computational process, and they often become unusable, thus requiring the program operator's intervention. Our work is aimed at eliminating these problems by implementing an ALE method based on the use of a fixed, structured or unstructured mesh that covers the complete (or maximum) domain occupied by the fluid at any time in the simulation and that remains fixed throughout the calculation. The moving interfaces are described using sets of marker points that define the different moving bodies or boundaries. The marker points can move freely over the basic mesh with a velocity that may be prescribed or be part of the calculated solution [7]. Figure 1 illustrates a mesh and interfaces configuration. At each time step in the calculation the elements intersected by one of the moving interfaces are subdivided to fit the boundary with a piecewise linear curve such that the computational nodes always remain on the elements sides. The modified mesh is used to calculate the flow in the portion of the domain occupied by fluid only once for that time and interface position. At the next time step the interfaces are advanced, the new intersections with the mesh are calculated, and a new local adaptation performed. Once the moving liquid-solid interface has gone through the stationary element, the element recovers its original form. Therefore, the mesh adaptation is performed only in those elements intersected by an interface and is local both in space and time. As a result, the method requires a minimal amount of interpolation and there are a fixed number of possible modifications to the intersected elements, three in the 2D case when quadrilateral elements are used and seven in the 3D case when hexahedrons are used. The situation is even simpler if the model is based on triangles in 2D and tetrahedrons in 3D, requiring only one kind of modification for 2D triangles and four in the case of four-node tetrahedrons. The use of this strategy eliminates the problem of maintaining the mesh quality and results in a robust formulation on arbitrary geometrical configurations.

At this time, the 2D four-node bilinear isoparametric element has been implemented to test the ideas, and the interface-moving algorithm has been combined with a first-order-in-time fractional step (projection) formulation [8] for incompressible flow. The accuracy of the model has been tested using an exact analytical solution to the case of 2D flow between two parallel plates separating with a prescribed velocity [9]. The error measures have been chosen by averaging over the computational nodes contained on a fixed a portion of the domain in order to obtain an estimate of the error at each time step, and then averaging those errors over a fixed-time interval. This results in one number representing the error in a computation for a specific spatial mesh. This process is repeated for two additional meshes, each refined so that the mesh size parameter is one half of the previous one, to estimate the convergence rate of the method as a function of mesh size. The results show the expected second-order convergence of the velocity and first-order convergence of the pressure as a function of the mesh size.

For more information contact David Carrington at dcarring@lanl.gov.

Funding Acknowledgments
DOE, Office of Energy Efficiency and Renewable Energy, Vehicle Technology Program

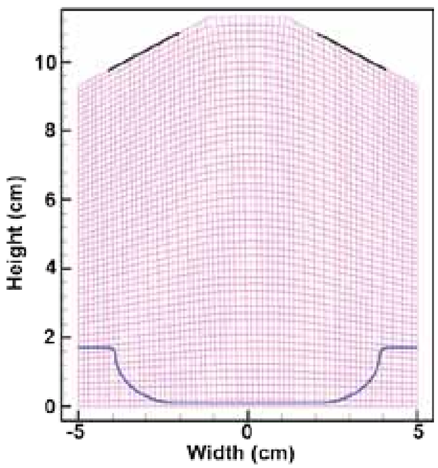

Fig. 2. Finite element mesh for the simulation of an engine piston chamber showing valve openings in black. The geometry and initial position of the top of the cylinder head is shown in blue at the bottom.

An example that illustrates the capability of the method to model the interface's motion is given in Fig. 2. This calculation has been designed to show the effectiveness of the method to model the geometrical changes due to the moving interfaces; it does not include combustion, but it does consider 2D laminar incompressible flow at low Reynolds number. The figure shows a 2D idealization of a piston chamber and a cylinder head that moves in the y direction according to $y(t) = y(0) + 2.5(1 + \sin(t - \pi/2))$, so the stroke of the piston is 5 cm. Figures 3 and 4 show the position of the piston, the flow field, and the pressure field at two instants; in Fig. 3 it is starting to move down after 3.5 seconds of simulation; after reaching its maximum height, the left (outlet) valve is closed and fluid enters the region through the right (inlet) valve. In Fig. 4 at 6.55 seconds the piston is initiating upwards motion and the flow leaves the domain through the right valve. The simulation was performed in 13,100 time steps of equal size to reach 6.55 seconds. The total CPU time required for this run in a 2009 Dell Optiplex 960 PC with a 3-MHz, ×86 Intel processor and 1.5 GB of physical memory is 3.79 minutes.

So far the present work has been restricted to laminar incompressible flow in two space dimensions in order to verify feasibility and accuracy of the method to model moving interfaces. The same methodology can be applied directly to high Reynolds number compressible and turbulent flows. The extension of the method to three space dimensions is currently under development.

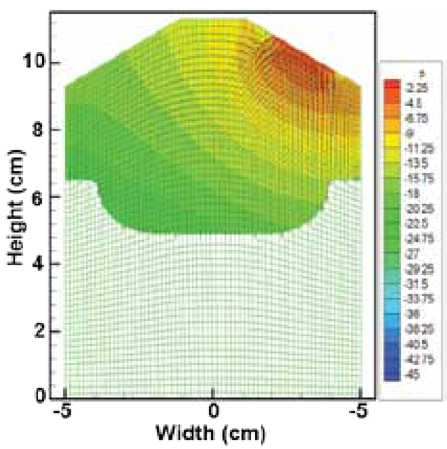

Fig. 3. Velocity and pressure after 3.5 seconds when the piston is moving back down and the inlet valve on the right-hand side is open. Maximum velocity is 4.4 cm/s.

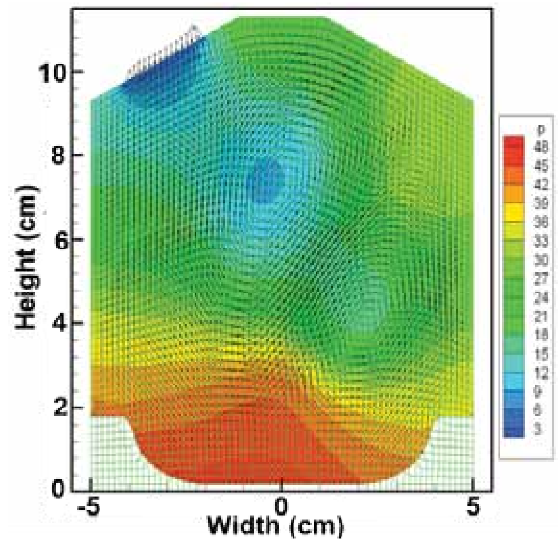

Fig. 4. Velocity and pressure after 6.55 seconds, the piston has just reversed its motion and is moving up. The inlet valve is closed and the outlet valve on the left is open. Maximum velocity is 7.7 cm/s.

[1] Torres, D.J. and M.F. Trujillo, *J Comp Phys* **219,** 943 (2006).

[2] Trescher, D., *Arch Comput Meth Eng* **15,** 67 (2008).

[3] Donea, J. et al., *Comput Meth Appl Mech Eng* **33,** 689 (1982).

[4] Farhat, C. and P. Geuzaine, *Comput Meth Appl Mech Eng* **193,** 4073 (2004).

[5] Saksono, P.H. et al., *Int J Numer Meth Eng* **71,** 1009 (2007).

[6] Löhner, R. and J.R. Cebral, *Arch Comput Meth Eng* **14,** 279 (2007).

[7] Zhao, P. and J.C. Heinrich, *J Comp Phys* **173,** 765 (2001).

[8] Quartapelle, L., *Numerical Solution of the Incompressible Navier-Stokes Equations,* Birkhäuser, Basel (1993).

[9] Drazin, P. and N. Riley, "The Navier-Stokes Equations: a Classification of Flows and Exact Solutions," *London Mathematical Society Lecture Series* **344,** Cambridge University Press, Cambridge (2006).